Theoretical Computer Science 19 (1982) 219–227
North-Holland Publishing Company

NOTE

2DST MAPPINGS ON LANGUAGES AND RELATED PROBLEMS*

Oscar H. IBARRA
Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, U.S.A.

Communicated by R. Book
Received May 1981
Revised October 1981

Abstract. Let $\mathscr{L}$ be a family of languages effectively closed under inverse homomorphism and intersection with regular sets and such that the languages have effectively constructible semilinear Parikh maps. We show that there is an algorithm to decide given a language L in $\mathscr{L}$ and a language R accepted by a one-way nondeterministic multicounter machine, where each counter makes exactly one reversal, whether $L \cap R$ is empty. This result has many applications. In particular, it can be used to show that there is an algorithm to decide given a language L in $\mathscr{L}$ and two-way deterministic sequential transducers (2DST's) S_1 and S_2 whether S_1 and S_2 are equivalent on L.

1. Introduction

The results in this paper were motivated by the following theorem and question posed in [8]:

Theorem. *It is decidable to determine given a context-free language L and deterministic generalized sequential machines (gsm's) S_1 and S_2 whether S_1 and S_2 are equivalent on L (i.e., $S_1(x) = S_2(x)$ for each x in L).*

Question. To what extent does the theorem above remain valid for other language families?

As far as we know, the theorem has not been shown for any other family of languages. In the special case when the gsm's are homorphisms, the following result is known [7]: The homomorphism equivalence problem is decidable for ETOL languages restricted to two-letter alphabets.

The gsm and homomorphism equivalence problems occur in many applications (see, e.g., [4, 5, 9]). These problems are undecidable for context-sensitive languages

* This research was supported in part by NSF Grant MCS78-01736.

[8], and hence, the only families of interest are the subfamilies of the context-sensitive family.

In this paper, we prove the following result which generalizes the theorem above: Let $\mathscr{L}$ be a family of languages effectively closed under inverse homomorphism and intersection with regular sets and such that the languages have effectively constructible semilinear Parikh maps. Then there is an algorithm to decide given a language L in $\mathscr{L}$ and two-way deterministic sequential transducers (with accepting states) S_1 and S_2 whether S_1 and S_2 are equivalent on L. Thus, the result holds for the language families listed below. The semilinearity of the Parikh maps are either easily shown or are proved in the references.

(1) State languages of degree $n \geq 1$ [22]. State languages of degree 1 are exactly the context-free languages. The "degree" defines a hierarchy between context-free and context-sensitive languages [22].

(2) Simple matrix languages of degree $n \geq 1$ [18]. Again, the degree defines a hierarchy between context-free and context-sensitive.

(3) Languages generated by absolutely parallel grammars [24].

(4) Languages generated by certain restricted forms of programmed grammars [25].

(5) Languages accepted by 1-way nondeterministic pushdown automata augmented by a finite number of 1-reversal counters [17].

(6) Languages accepted by 2-way nondeterministic finite automata (with endmarkers) augmented by a finite number of 1-reversal counters, where in every accepting computation, the input head visits each position on the input tape no more than a fixed number of times [16].

(7) Languages accepted by nondeterministic on-line Turing machines with one working tape preset to some string in L, where L is in a full semiAFL whose languages have effectively constructible semilinear Parikh maps. The worktape head is "finite-visit" in that it can visit a square no more than a fixed number of times [13].

(8) Languages accepted by finite-turn checking automata [13, 27].

(9) Languages accepted by controlled pushdown automata [19].

In addition to the families above, new families with the effectively constructible semilinear Parikh map property can be obtained using certain operations on sets, e.g., AFL operations, replications, commutative closure, etc. [11]. These new families will also have a decidable 2-way transducer equivalence problem.

We conclude this section with some definitions and notation.

Definition. Let N denote the set of nonnegative integers and let N^r be the cartesian product of N with itself r times. A subset Q of N^r is called a *linear set* if there exist $v_0, v_1, \ldots, v_m$ in N^r such that $Q = \{v \mid v = v_0 + k_1 v_1 + \cdots + k_m v_m$, each k_i in $N\}$. $v_0, v_1, \ldots, v_m$ are called the generators of Q. Any finite union of linear sets is called a *semilinear set.*

Definition. Let Σ be a finite alphabet and let $\alpha = \langle a_1, a_2, \ldots, a_r \rangle$ list the elements of Σ in some order. For x in Σ^*, define the r-tuple of nonnegative integers $f_\alpha(x) = (\#a_1(x), \#a_2(x), \ldots, \#a_r(x))$, where $\#a_i(x)$ is the number of occurrences of symbol a_i in x. For $L \subseteq \Sigma^*$, the mapping $f_\alpha(L) = \{f_\alpha(x) | x \text{ in } L\}$ is called a *Parikh* map of L. The languages in a family $\mathscr{L}$ have effectively constructible semilinear Parikh maps if for each L in $\mathscr{L}$, $f_\alpha(L)$ is a semilinear set effectively constructible from L.

Notation. Throughout the paper, $\mathbb{C}$ denotes the class of 1-way nondeterministic multicounter machines, where each counter makes exactly 1 reversal. For a machine M in $\mathbb{C}$, $T(M)$ denotes the language accepted by M. We assume without loss of generality that the counters are zero on acceptance. The class $\mathbb{C}$ has been studied in a number of papers, e.g., in [1, 16, 17].

2. Decision problems concerning $\mathbb{C}$

We begin with the following theorem which is a generalization of a result in [17]. In [17], the result is shown only for the family of context-free languages.

Theorem 1. *Let $\mathscr{L}$ be a family of languages effectively closed under inverse homomorphism and intersection with regular sets and such that the languages have effectively constructible semilinear Parikh maps. Then for each L in $\mathscr{L}$ and each M in $\mathbb{C}$, $L \cap T(M)$ has an effectively constructible semilinear Parikh map. Hence, there is an algorithm to decide whether $L \cap T(M)$ is empty (respectively, infinite).*

Proof. Let $L \subseteq \Sigma^*$ be in $\mathscr{L}$ and M be a 1-way nondeterministic machine with k 1-reversal counters. We shall show that $L \cap T(M)$ has an effectively constructible semilinear Parikh map. Since it is trivial to check whether a semilinear set is empty or infinite the theorem then follows.

Let $\Sigma = \{a_1, \ldots, a_r\}$ be the input alphabet of M.[1] We first construct a regular set R_M which consists of encodings of all "possible" accepting computations of M on inputs in Σ^*. $R_M \subseteq \Delta^*$, where Δ contains Σ, new symbols $\¢_1, \ldots, \¢_k, \$_1, \ldots, \$_k$, as well as other symbols representing the transition rules of M. A string $\alpha_1 \ldots \alpha_m$ is in R_M if and only if $\alpha_1 \ldots \alpha_m$ represents a possible accepting computation of M on some input x. Thus, $\alpha_1 \ldots \alpha_m$ encodes x as well as the rules used by M on its computation on x. The action of M on the ith counter, e.g., +1 or −1 is represented in the string $\alpha_1 \ldots \alpha_m$ by an occurrence of $\¢_i$ or $\$_i$, respectively. Thus, the number of occurrences of $\¢_i$ in $\alpha_1 \ldots \alpha_m$ represents the largest integer stored in counter i during the computation of M on x. The details of the construction of a finite automaton accepting R_M is described in [20]. Now define a homomorphism h from

[1] There is no loss of generality in assuming that M has input alphabet Σ since $\mathscr{L}$ is effectively closed under intersection with regular sets.

Δ^* to Σ^* so that $h(\alpha_1 \ldots \alpha_m) = x$. Then the language $L' = h^{-1}(L) \cap R_M$ is in $\mathcal{L}$, and L' has the property that $\alpha_1 \ldots \alpha_m$ is in L' if and only if $\alpha_1 \ldots \alpha_m$ represents a possible accepting computation of M on $x = h(\alpha_1 \ldots \alpha_m)$ in L. It follows that $L \cap T(M) \neq \emptyset$ if and only if there is a string $\alpha_1 \ldots \alpha_m$ in L' such that for $1 \leq i \leq k$, the number of occurrences of $\$_i$ (increments to counter i) = the number of occurrences of $\$_i$ (decrements to counter i). Since L' is in $\mathcal{L}$, $Q_1 = f_\beta(L')$ is an effectively constructible semilinear set, where $\beta = \langle a_1, \ldots, a_r, \$_1, \$_1, \ldots, \$_k, \$_k, b_1, \ldots, b_s \rangle$, $b_1, \ldots, b_s$ being the other symbols in $\Delta - \Sigma - \{\$_1, \$_1, \ldots, \$_k, \$_k\}$. Let $Q_2 = \{(l_1, l_2, \ldots, l_r, i_1, j_1, \ldots, i_k, j_k, t_1, t_2, \ldots, t_s) \mid l_1, \ldots, l_r \geq 0, i_1 = j_1 \geq 0, \ldots, i_k = j_k \geq 0, t_1, \ldots, t_s \geq 0\}$. Clearly, Q_2 is a semilinear set. Then $Q_3 = Q_1 \cap Q_2$ is a semilinear set effectively constructible from Q_1 and Q_2 [10]. Let Q_4 be the semilinear set obtained from Q_3 by deleting the last $2k+s$ coordinates from the generators of the linear sets forming Q_3. Then $Q_4 = f_\alpha(L \cap T(M))$, where $\alpha = \langle a_1, \ldots, a_r \rangle$, completing the proof. □

The proof of the next lemma is straightforward and therefore omitted.

Lemma 1. *Let M be in $\mathbb{C}$. We can effectively construct a machine M' in $\mathbb{C}$ such that $T(M') = T(M)$ and each x in $T(M')$ has an accepting computation with the following properties:*

(1) *For some positive integer v (which depends on x) all counters reach the value v for their maximum;*

(2) *On each atomic move, at least 1 counter changes value.*

Using Lemma 1 and a construction similar to that of Theorem 1, we can prove the following result. We omit the proof and refer the reader to [20] for details.

Theorem 2. *Let $\mathcal{L}$ be a family of languages effectively closed under inverse homomorphism and intersection with regular sets. Then* (1) *is decidable if and only if* (2) *is decidable:*

(1) *Given a language L in $\mathcal{L}$ and a machine M in $\mathbb{C}$, is the set $L \cap T(M)$ empty?*

(2) *Given a language L in $\mathcal{L}$ over the alphabet Σ, is the set $E(L) = \{x \mid x$ in L and for all a, b in Σ, the number of a's in x = the number of b's in $x\}$ empty?*

From Theorems 1 and 2, we have

Corollary 1. *Problems* (1) *and* (2) *of Theorem 2 are decidable if the languages in $\mathcal{L}$ have effectively constructible semilinear Parikh maps.*

The next theorem shows that the semilinearity requirement in Corollary 1 cannot be removed.

Theorem 3. *Problems* (1) *and* (2) *of Theorem 2 are undecidable for any family* $\mathscr{L}$ *containing the language* $A = \{0^n 1^{n^2} \mid n \geq 1\}^+$. *The result also holds when* $\mathscr{L}$ *contains* $B = \{1^{n^2} 0^n \mid n \geq 1\}^+$ *instead of* A.

Proof. By Theorem 2, we need only show the undecidability of problem (1). The proof uses the undecidability of Hilbert's tenth problem [23], i.e., the problem of deciding given a polynomial $p(x_1, \ldots, x_t)$ with integer coefficients whether $p(x_1, \ldots, x_t) = 0$ has a positive integer solution.

We describe an algorithm which constructs for any given polynomial $p(x_1, \ldots, x_t)$, a language R_p accepted by a 1-way machine with 1-reversal counters such that $A \cap R_p$ is nonempty if and only if $p(x_1, \ldots, x_t) = 0$ has a positive integer solution. We illustrate the construction via an example. The generalization is straightforward. Suppose $p(x, y) = xy - 2x^2 + y - 5$. Let

$$R_p = \{0^x 1^{v_1} 0^y 1^{v_2} 0^{x+y} 1^{v_3} \mid x, y, v_1, v_2, v_3 \geq 1, \lfloor (v_3 - v_1 - v_2)/2 \rfloor - 2v_1 + y - 5 = 0\}.$$

Clearly, R_p can be accepted by a 1-way machine with 1-reversal counters, in fact, deterministically. Then

$$A \cap R_p = \{0^x 1^{x^2} 0^y 1^{y^2} 1^{x+y} 1^{(x+y)^2} \mid x, y \geq 1, xy - 2x^2 + y - 5 = 0\}.$$

Hence, $A \cap R_p$ is nonempty if and only if $p(x, y) = xy - 2x^2 + y - 5 = 0$ has a positive integer solution. The result now follows from the undecidability of Hilbert's tenth problem. The proof for the case when $\mathscr{L}$ contains B instead of A is similar. □

We can also prove the following

Theorem 4. *Problems* (1) *and* (2) *of Theorem 2 are undecidable for any family* $\mathscr{L}$ *containing the language* $C = \{(0^i 1)^j \# \mid i, j \geq 1\}^+$. *The result also holds when* $\mathscr{L}$ *contains* $D = \{0^{i_1} \# 0^{i_2} \# \ldots 0^{i_k} \# (0^{i_1} 1)^{j_1} \# (0^{i_2} 1)^{j_2} \# \ldots (0^{i_k} 1)^{j_k} \# \mid k \geq 1, i_1, i_2, \ldots, i_k, j_1, j_2, \ldots, j_k \geq 1\}$ *instead of* C.

Proof. The proof is similar to that of Theorem 3. For the polynomial $p(x, y) = xy - 2x^2 + y - 5$, let $R_p = \{0^x 10^{i_1} 1 \ldots 0^{i_{x-1}} 1 \# 0^y 10^{j_1} 1 \ldots 0^{j_{y-1}} 1 \# 0^{x+y} 10^{k_1} 1 \ldots 0^{k_{x+y-1}} 1 \# \mid x, y, i_1, \ldots, i_{x-1}, j_1, \ldots, j_{y-1}, k_1, \ldots, k_{x+y-1} \geq 1, \lfloor (w - u - v)/2 \rfloor - 2u + y - 5 = 0$, where $u = x + i_1 + \cdots + i_{x-1}$, $v = y + j_1 + \cdots + j_{y-1}$, $w = x + y + k_1 + \cdots + k_{x+y-1}\}$. Then $C \cap R_p = \{(0^x 1)^x \# (0^y 1)^y \# (0^{x+y} 1)^{x+y} \# \mid x, y \geq 1, xy - 2x^2 + y - 5 = 0\}$ is nonempty if and only if $p(x, y)$ has a positive integer solution. The proof for the case when $\mathscr{L}$ contains D instead of C is similar. □

Corollary 2. *Problems* (1) *and* (2) *of Theorem 2 are undecidable for the family* $\mathscr{L}_{\text{DCSA}}$ *of languages accepted by 1-way deterministic checking stack automata.*[2]

[2] A "checking stack" [12] is a pushdown store which cannot be erased but can be entered in a read-only mode. Moreover, once the stack is entered, it can no longer be written upon.

Hence, it is undecidable to determine for an arbitrary language $L \subseteq \Sigma^*$ in $\mathscr{L}_{\text{DCSA}}$ whether there is an x in L such that for all a, b in Σ, the number of a's in x = the number of b's in x.

Proof. The language D (defined in Theorem 4) is in $\mathscr{L}_{\text{DCSA}}$. □

3. Applications of Theorem 1

Theorem 1 can be used to prove the decidability of certain questions concerning mappings of languages. We shall give three examples. The first generalizes the main result in [8].

Definition. A 2-way deterministic sequential transducer with accepting states (2DST), S, is a 2-way deterministic finite automaton with a 1-way write-only output tape. The input is provided with endmarkers ¢ and \$, and on each move, S can write a (possibly null) string. S defines a mapping as follows: For each input string x,

$$S(x) = \begin{cases} y, & \text{if } S \text{ when given } ¢x\$ \text{ halts in an accepting state with output } y, \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

Two 2DST's S_1 and S_2 are equivalent on a language L if for every x in L, $S_1(x)$ and $S_2(x)$ are defined and are equal or they are both undefined. Note that a generalized sequential machine (gsm) is a special case of a 2DST.

In [15] an algorithm is given which constructs for any 2DST's S_1 and S_2 over an input alphabet Σ a 1-way nondeterministic machine M with 2 1-reversal counters such that $T(M)$ is empty if and only if S_1 and S_2 are equivalent on Σ^*. That algorithm combined with Theorem 1 gives us the following generalization of the main result in [8].

Theorem 5. *Let $\mathscr{L}$ be a family of languages effectively closed under inverse homomorphism and intersection with regular sets and such that the languages have effectively constructible semilinear Parikh maps. Then there is an algorithm to decide given a language L in $\mathscr{L}$ and 2DST's S_1 and S_2 whether S_1 and S_2 are equivalent on L.*

The notion of "balance" of homomorphisms has been found very useful in studying decidability and undecidability of certain questions concerning formal languages as well as in characterizing some complexity classes of languages [2, 3, 5, 6, 26]. Let h_1 and h_2 be homomorphisms on Σ^* and $L \subseteq \Sigma^*$. Let k be a nonnegative integer. (h_1, h_2) has k-balance on L if $\text{bal}(y) = \text{abs}(|h_1(y)| - |h_2(y)|) \leq k$ for each

prefix y of every string in L.[3] We can generalize the definition to gsm's as follows. A pair (S_1, S_2) of gsm's (with accepting states) has k-balance on a language L if the following holds for every x in L:

(1) $S_1(x)$ is defined if and only if $S_2(x)$ is defined;

(2) If $S_1(x)$ and $S_2(x)$ are defined, then for every prefix y of x, $\text{abs}(|\hat{S}_1(y)| - |\hat{S}_2(y)|) \leq k$, where $\hat{S}_i(y)$ denotes the prefix of output $S_i(x)$ due to y.

Theorem 6. *Let $\mathscr{L}$ be as in Theorem 5. Then there is an algorithm to decide given L in $\mathscr{L}$ and gsm's S_1 and S_2 whether (S_1, S_2) has k-balance on L for some $k \geq 0$. Moreover, in the positive case, the unique minimal value of k can effectively be obtained.*

Proof. Let L be in $\mathscr{L}$ and S_1 and S_2 be two gsm's. Assume that Σ is the common alphabet of L, S_1 and S_2. Let $\#$ be a new symbol not in Σ. Define a language R consisting of all strings $x\#^k$, where x is in Σ^*, $k \geq 0$, $S_1(x)$ is defined if and only if $S_2(x)$ is defined, and if they are defined, for some prefix y of x, $\text{abs}(|\hat{S}_1(y)| - |\hat{S}_2(y)|) = k$. Clearly, R can be accepted by a 1-way nondeterministic machine with 2 1-reversal counters. Define a homomorphism $h: (\Sigma \cup \{\#\})^* \to \Sigma^*$ by: $h(\#) = \varepsilon$ and $h(a) = a$ for each a in Σ. Then $L' = h^{-1}(L)$ is in $\mathscr{L}$ and, by Theorem 1, $L' \cap R$ has an effectively constructible semilinear Parikh map, Q. Delete all the components except that which corresponds to the symbol $\#$ from the generators of the linear sets of Q. Let Q' be the resulting semilinear set. Then (S_1, S_2) has k-balance on L for some $k \geq 0$ if and only if Q' is finite. Moreover, if Q' is finite, then the minimal such k is equal to the largest nonnegative integer in Q'. □

As a final example, we prove a generalization of a result in [21] (see also [14]).

Theorem 7. *Let $\mathscr{L}$ be as in Theorem 5. Then there is an algorithm to decide given a language $L \subseteq \Sigma^*$ in $\mathscr{L}$ and n-tuples $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ of nonnull strings in Σ^+ whether there exists a string w in L with the following property:*

$$w = x_{i_1} \ldots x_{i_k} = y_{j_1} \ldots y_{j_k} \text{ for some } k \geq 1,\ 1 \leq i_1, \ldots, i_k, j_1, \ldots, j_k \leq n, \text{ and } (i_1, \ldots, i_k) \text{ is a permutation of } (j_1, \ldots, j_k). \quad (*)$$

Proof. Given $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$, define the language $R = \{w \mid w$ in Σ^+, w satisfies property $(*)\}$. It is easy to construct a 1-way nondeterministic machine with $2n$ 1-reversal counters accepting R—a counter c_i (respectively, d_i) is used to keep track of the number of times x_i (respectively, y_i) is used in w. The result follows from Theorem 1. □

[3] abs is abbreviation for absolute value; $|h_i(y)|$ denotes the length of $h_i(y)$.

The semilinearity requirement on $\mathscr{L}$ in Theorem 7 is necessary. As a counter-example, suppose $\mathscr{L}$ contains one of the languages A, B, C, D defined in Theorems 3 and 4. Clearly, for $L \subseteq \{a_1, \ldots, a_n\}^*$ in $\mathscr{L}$, a string w in L satisfies (*) of Theorem 7 for $(x_1, \ldots, x_n) = (a_1, \ldots, a_n)$ and $(y_1, \ldots, y_n) = (a_2, \ldots, a_n, a_1)$ if and only if the number of occurrences of a_i in w = the number of occurrences of a_j in w for all $i \neq j$. The undecidability then follows from Theorems 2–4.

References

[1] B. Baker and R.V. Book, Reversal-bounded multipushdown machines, *J. Comput System Sci.* **8** (1974) 315–332.

[2] R.V. Book and F.J. Brandenburg, Equality sets and complexity classes, *SICOMP* **9** (1980) 729–743.

[3] K. Culik II, A purely homomorphic characterization of recursively enumerable sets, *J. ACM* **26** (1979) 345–350.

[4] K. Culik II, Some decidability results about regular and pushdown translations, *Information Processing Lett.* **8** (1979) 5–8.

[5] K. Culik II, Homomorphisms: decidability, equality and test sets, *Proc. International Symposium on Formal Language Theory*, Santa Barbara, CA (1979).

[6] K. Culik II and N.D. Diamond, A homomorphic characterization of time and space complexity classes of languages, *Internat. J. Comput. Math.* **8** (1980) 207–222.

[7] K. Culik II and J.L. Richier, Homomorphism equivalence on ETOL Languages, *Internat. J. Comput. Math.* **7** (1979) 43–51.

[8] K. Culik II and A. Salomaa, On the decidability of homomorphism equivalence for languages. *J. Comput. System Sci.* **17** (1978) 163–175.

[9] K. Culik II and A. Salomaa, Test sets and checking words for homomorphism equivalence, *J. Comput. System Sci.* **20** (1980) 379–395.

[10] S. Ginsburg and E. Spanier, Bounded Algol-like Languages, *Trans. Amer. Math. Soc.* **113** (1964) 333–368.

[11] S. Ginsburg and E. Spanier, AFL with the semilinear property, *J. Comput. System Sci.* **5** (1971) 365–396.

[12] S.A. Greibach, Checking automata and one-way stack languages, *J. Comput. System Sci.* **3** (1969) 196–217.

[13] S.A. Greibach, One-way finite visit automata, *Theor. Comput. Sci.* **6** (1978) 175–221.

[14] S.A. Greibach, A remark on code sets and context-free languages, *IEEE Trans. Comput.* (1975) 741–742.

[15] E.M. Gurari, The equivalence problem for deterministic two-way sequential transducers is decidable, *Proc. 21st Symposium on Foundations of Computer Science* (1980) 83–85.

[16] E.M. Gurari and O.H. Ibarra, The complexity of decision problems for finite-turn multicounter machines, *J. Comput. System Sci.* **22** (1981) 220–229.

[17] O.H. Ibarra, Reversal-bounded multicounter machines and their decision problems, *J. ACM* **25** (1978) 116–133.

[18] O.H. Ibarra, Simple matrix languages, *Information and Control* **17** (1970) 359–394.

[19] O.H. Ibarra, Controlled pushdown automata, *Information Sci.* **6** (1973) 327–342.

[20] O.H. Ibarra, Some decision problems and applications, University of Minnesota, Department of Computer Science, Technical Report 81-12 (1981).

[21] O.H. Ibarra and C.E. Kim, A useful device for showing the solvability of some decision problems, *J. Comput. System Sci.* **13** (1976) 153–160.

[22] T. Kasai, An hierarchy between context-free and context-sensitive languages, *J. Comput. System Sci.* **4** (1970) 492–508.

[23] Y. Matijasevic, Enumerable sets are Diophantine, *Soviet Math. Dokl.* **11** (1970) 354–357.

[24] V. Rajlich, Absolutely parallel grammars and two-way finite state transducers, *J. Comput. System Sci.* **6** (1972) 324–342.
[25] D.J. Rosenkrantz, Programmed grammars and classes of formal languages, *J. ACM* **16** (1969) 107–131.
[26] A. Salomaa, Equality sets for homomorphisms of free monoids, *Acta Cybernetica* **4** (1978) 127–139.
[27] R. Siromoney, Finite-turn checking automata, *J. Comput. System Sci.* **5** (1971) 549–559.